



Üldinfo

Kõigis ülesannetes:

- Piirangud on kirjas võistlussüsteemi lehel "Ülevaade"
- Võistluse serverist saad abifailide paki, milles on nii hindamisprogrammi kui ka lahendusfailide näited, näidissisendid ning kompileerimis- ja käitamiskriptid.
- Sa võid igale ülesandele lahendusi esitada kuni 50 korda. Iga esitus peab olema ühes failis.
- Oma lahendust näidishindajaga testides pead kasutama sisendit, mis vastab ülesande tekstis kirjeldatud vormingule ja piirangutele. Vastasel juhul on hindamisprogrammi käitumine ennustamatu.
- Näidishindaja sisendis peab iga kahe samal real kõrvuti oleva lekseemi (sõna või arvu) vahel olema üks tühik, kui ülesande tekstis pole öeldud teisiti.
- Oma lahendusi oma arvutis testides soovitame kasutada abifailide pakis olevaid skripte. Pane tähele, et kasutame kompilaatori käsureal `-std=gnu++17` võtit.
- Kui sa ei saa lahendusi CMS liidese kaudu esitada, siis võid kasutada `ioisubmit` käsku, et saata oma lahendused hindamiseks pärast võistluse lõppu.
 - Käivita `ioisubmit <ülesande_lühinimi> <lähtekoodi_fail>` kaustas, kus asub `<lähtekoodi_fail>`.
 - Palu klassivalvuril teha käsu `ioisubmit` väljundist pilt ja see korraldajatele saata. Kui see samm tegemata jääb, siis sinu lahendust ei hinnata.
 - Kui võistled kaugteel, siis palu oma valvuril teha käsu `ioisubmit` väljundist pilt ja saata see korraldajatele.

Tähistused

Ülesannete tekstides on funktsioonide signatuurid antud abstraktsete tüüpidega `void`, `bool`, `int`, `int[]` (massiiv) ja `union(bool, int[])`.

Keeles C++ kasutavad hindamiskriptid järgmisi andmetüüpe või realisatsioone:

| | | | |
|-------------------|-------------------|------------------|-------------------------------------|
| <code>void</code> | <code>bool</code> | <code>int</code> | <code>int[]</code> |
| <code>void</code> | <code>bool</code> | <code>int</code> | <code>std::vector<int></code> |

| | |
|---|-----------------------|
| <code>union(bool, int[])</code> | massiivi a pikkus |
| <code>std::variant<bool, std::vector<int>></code> | <code>a.size()</code> |

Keeles C++ on `std::variant` defineeritud päises `<variant>`. Meetod, mis tagastab tüüpi `std::variant<bool, std::vector<int>>` võib tagastada kas `bool` või `std::vector<int>` tüüpi väärtuse. Allpool toodud kood illustreerib kolme võimalikku näidet, kuidas funktsioon võib tagastada `std::variant` tüüpi väärtuse.

```
std::variant<bool, std::vector<int>> foo(int N) {
    return N % 2 == 0;
}
std::variant<bool, std::vector<int>> goo(int N) {
    return std::vector<int>(N, 0);
}
std::variant<bool, std::vector<int>> hoo(int N) {
    if (N % 2 == 0) {
        return false;
    }
    return std::vector<int>(N, 0);
}
```